Beyond Bit Matching: Orthogonal Watermarks for Collusion-Resistant Image Fingerprinting

Zhakshylyk Nurlanov Tobias Weißberg Florian Bernard
University of Bonn Lamarr Institute for Machine Learning and Artificial Intelligence

Ten colluders average their copies. VectorMark still names all ten.

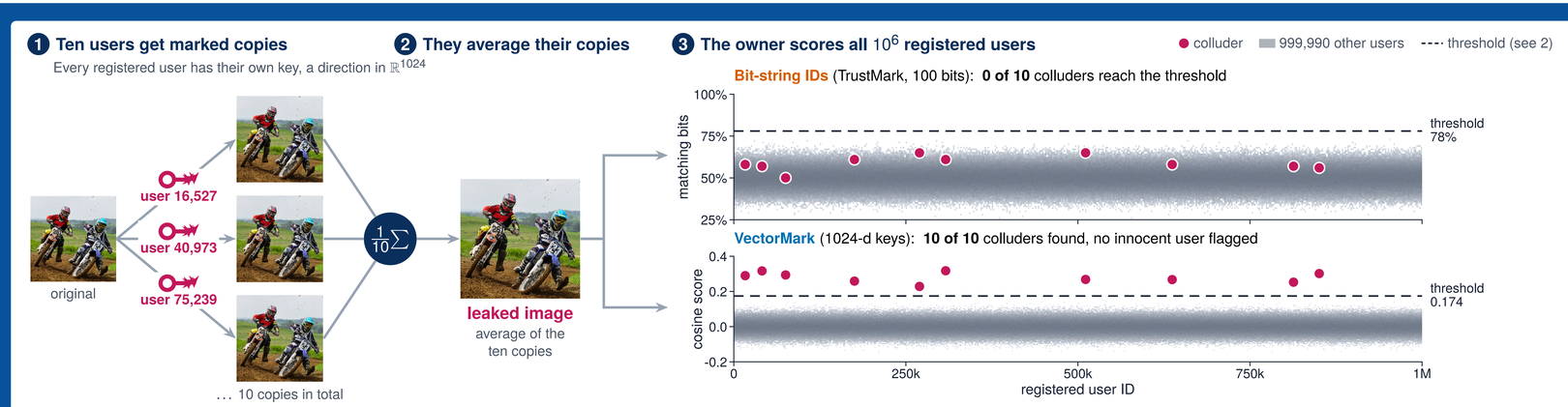
Each user's copy carries an invisible key: a direction in a 1024-dimensional space. The average of several copies still points toward every colluder. So scoring all registered users finds the whole group.

98.0% of trials find exactly the ten colluders among 10^6 users

1.1% of trials flag an innocent user (target: 1%)

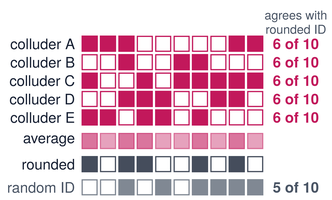
40.3 dB average PSNR of the watermarked images

nurlanov.me/vectormark



1 Fingerprinting and averaging collusion

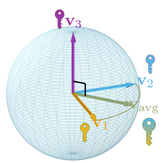
Fingerprinting. Each recipient of an image gets a personal copy. The copy hides the recipient's ID as an invisible watermark. If a copy leaks, the ID shows who received it. Typical uses: licensed photos, press previews, confidential designs.



Averaging collusion. Several recipients can team up and average their copies pixel by pixel. We call them colluders. The image still looks clean, but each ID is now much weaker.

Several labels on purpose. VectorMark can also write several labels into one image, such as creator, licence and distributor. For ten labels, a search over 10^6 registered labels returns exactly those ten in 99% of images.

2 User keys and detection thresholds



Keys as directions. VectorMark turns each user ID into a direction, a unit vector $\mathbf{v}$ in $\mathbb{R}^{1024}$. In so many dimensions, two random directions are almost perpendicular. So the average of K colluders' keys keeps the same cosine with each of them:

$$\left\langle \frac{\sum_{i=1}^{K} \mathbf{v}_i}{\left\| \sum_{i=1}^{K} \mathbf{v}_i \right\|}, \mathbf{v}_j \right\rangle = \frac{1}{\sqrt{K}}$$

0.32 for ten colluders; an unrelated key scores about 0

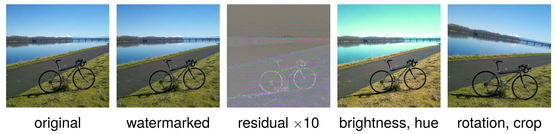
Thresholds. A user is reported when their score passes a threshold τ. We pick τ so that an unrelated user passes it only with a small probability α. With N users and a 1% false-alarm budget in total, $\alpha = 1\%/N$.

key	how an unrelated user scores	$\alpha = 10^{-6}$ per user	$\alpha = 10^{-8}$ 1% over 10^6 users
100-bit ID		74.0%	78.0%
256-bit ID	matching bits $\sim$ Binomial$(L, \frac{1}{2})$	65.2%	67.6%
1024-bit ID		57.5%	58.9%
1024-d direction	cosine S: $\frac{1+S}{2} \sim \text{Beta}(\frac{D-1}{2}, \frac{D-1}{2})$	**0.148**	**0.174**

For 10^6 users, $\alpha = 10^{-8}$: this gives the dashed lines at the top (78% of 100 bits, cosine 0.174). Panel 4 and the ablation use $\alpha = 10^{-6}$. Longer IDs need a smaller share of matching bits.

3 Architecture and training





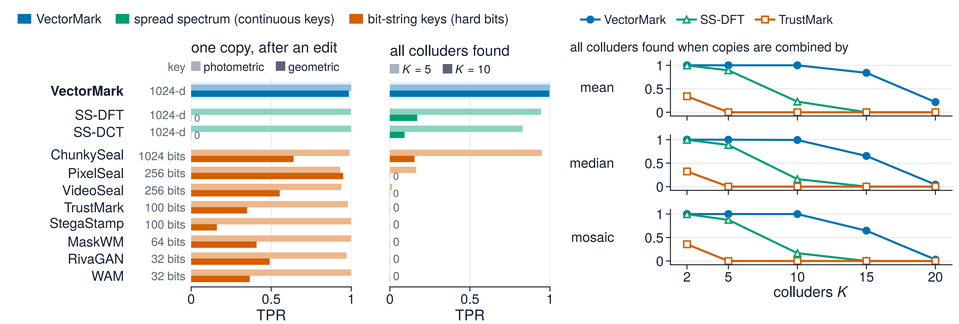
Encoder E hides the watermark, and extractor F reads the key back. A JND mask keeps it in textured areas, where it is hard to see. Training uses random edits, like the last two images. The score below each edited image is the cosine between the key read from that image and the user's key. Above 0.148, the user is found.

Collusion loss

$$\mathcal{L}_{\text{col}} = \underbrace{\left\| \bar{I} - E(I, \bar{\mathbf{v}}) \right\|^2}_{\text{looks like one copy marked with } \bar{\mathbf{v}}} + \underbrace{\left[1 - \cos(F(T(\bar{I})), \bar{\mathbf{v}})\right] + \left[1 - \cos(F(T(E(I, \bar{\mathbf{v}}))), \bar{\mathbf{v}})\right]}_{\text{the extractor reads } \bar{\mathbf{v}} \text{ from both, even after an edit } T}$$

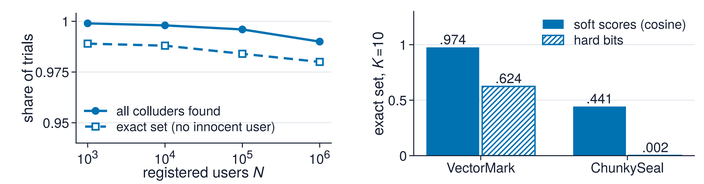
$\bar{I} = \frac{1}{K} \sum_k E(I, \mathbf{v}_k)$ is the average of K marked copies, and $\bar{\mathbf{v}}$ is the average of their keys, scaled to length 1.

4 Robustness to edits and to collusion, compared with ten methods



False-alarm rate $\alpha = 10^{-6}$ per user (see 2). Bit-string methods are scored by matching hard bits, continuous keys by cosine. A collusion test passes only if all K colluders are found. Median and mosaic (right) were not used in training.

5 Registry size and ablation studies

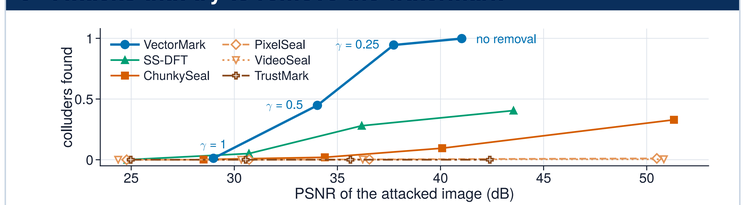


Left: ten colluders, every registered user scored, 1% false-alarm budget. Right: the same watermarked images at 10^6 users, scored by cosine or by rounded bits. Exact set: all colluders found and no one else. This is stricter than panel 4 ($\alpha = 10^{-8}$).

colluders K	τ	2	5	10	15	20
$D = 100$	0.453	1.00	.008	.000	.000	.000
$D = 256$	0.291	1.00	.985	.007	.000	.000
$D = 576$	0.196	1.00	1.00	.988	.321	.001
$D = 1024$	0.148	1.00	1.00	1.00	.977	.650
$D = 1024$, no collusion loss	0.148	1.00	1.00	1.00	.927	.234

Ablation study: all colluders found, $\alpha = 10^{-6}$. More dimensions give a lower threshold τ, so more colluders fit above it. The collusion loss helps most at 15 to 20 colluders.

6 Attacks that try to remove the watermark



Subspace removal: the attacker estimates the watermark directions from the copies and subtracts them, with strength γ. Same attack for all methods; ten colluders, 10^6 users.

Attack on VectorMark, $N = 10^6$	K	Exact set	Recall	Precision	PSNR
Averaging only	10	.979	.998	.999	41.03
WmForger	5	.725	.838	.904	32.68
Regen-VAE	5	.980	.998	.998	36.81
SD1.5 img2img	5	.000	.000	.000	25.50

When every user's key is a direction in a high-dimensional space, all colluders stay traceable after averaging, with a false-alarm rate fixed in advance.

Methods: Cox et al. 1997 (SS-DCT, SS-DFT); Tancik et al. 2020 (StegaStamp); Zhang et al. 2019 (RivaGAN); Bui et al. 2025 (TrustMark); Hu et al. 2025 (MaskWM); Sander et al. 2025 (WAM); Fernandez et al. 2024 (VideoSeal); Souček et al. 2025 (PixelSeal, WmForger); Petrov et al. 2025 (ChunkySeal); Zhao et al. 2024 (Regen-VAE); Rombach et al. 2022 (SD1.5); Wu et al. 2017 (JND). **Funding:** ERC Starting Grant no. 101160648.

Funded by the European Union